

TURBO CODES IN AMC SYSTEMS FOR BLIND IDENTIFICATION

VELAGA SYAMYA JAYASAI¹ & CH.RAVI KUMAR²

¹PG Scholar, Department of ECE, SIR C.R..Reddy College of Engineering, Eluru, Andhra Pradesh, India

²Sr. Assistant Professor, Department of ECE, SIR C.R..Reddy College of Engineering, Eluru, Andhra Pradesh, India

ABSTRACT

Adaptive modulation and coding (AMC) systems are essential in Blind identification for channel codes. Since Turbo codes are popular in AMC systems, it's necessary to identify its parameters. In this paper, we focus on the identification for Turbo codes from a closed-set. The proposed approach firstly identifies the first component code by accumulating Log-Likelihood Ratio (LLR) for syndrome a posteriori probability, then the interleaver and the other component code are identified by decoding based on zero insertion and LLR accumulation. This approach is robust to noise due to LLR. Moreover, it applies to both symmetric Turbo codes with two same component codes and asymmetric Turbo codes with two different component codes. Simulation results demonstrate that the proposed blind identification scheme is able to identify Turbo codes at signal-to-noise ratio (SNR) larger than 3.5dB. The improvement for identification of RSC2 and the application for Turbo codes with various rates. we design a novel blind encoder parameter estimator for turbo codes. In this scheme, we propose to separate the feedback components from the forward path in a recursive convolution encoder so as to blindly estimate the parameters. The simulations demonstrate the average estimation performance can lead to more than 95% accuracy for the channel condition with the signal-to noise ratio at 5 dB. Once the encoder parameters are blindly estimated, the corresponding decoder can be implemented to retrieve the original information sequences.

KEYWORDS: Blind Identification; Turbo Codes; zero Insertion Decoding; LLR

Received: Sep 07, 2017; **Accepted:** Oct 25, 2017; **Published:** Dec 07, 2017; **Paper Id.:** IJEIERDDEC20171

I. INTRODUCTION

Adaptive modulation and coding (AMC) is used to meet high demands on throughput in modern wireless systems. The idea is to change the modulation format and coding rate on the fly in order to adapt to a changing channel quality. Adaptive modulation and coding (AMC) techniques can dynamically select an appropriate combination of modulator and channel encoder from a candidate pool to adjust the time varying of the channel [1]-[3]. This adaptation mechanism is typically supported by a control channel on which the currently used modulation and coding parameters are signaled. These control channels themselves are quite information heavy in a modern wireless system and would benefit from the use of AMC in order to conserve channel resources in favor of payload data. Then in principle the modulation and coding parameters of these control channels would need to be sent over a "control-channel for the control channel", and so on. However, instead of explicitly signaling the AMC parameters, one can use so-called blind decoding.

Apparently, the receiver must synchronize the changes in modulation or coding mechanisms at the transmitter. Thus it is important for the receiver to confirm or to identify the parameters of channel codes.

Turbo codes are very popular as its near-Shannon-limit performance [4]. They have been widely applied

to many communication systems, such as deep space communication systems and mobile communication systems [5]-[6]. Turbo codes are also important alternative channel coding to AMC systems.

Recently many researchers dedicated themselves to blind identification for Turbo codes [7]-[12]. Blind estimation algorithm in [7] estimated the parameters of Turbo codes, e.g., code-word length, code rate, length of interleaver based on Gauss-Jordan matrix transformation. In [8], expectation-maximization (EM) algorithm was introduced to estimate the component code encoder. Reconstruction of Turbo codes based on searching dual words of low weight was presented in [9]. Most of the existing methods are sensitive to errors as they are mainly based on solving equations of hard-decision of the demodulator output sequence. And they mostly deal with symmetric Turbo codes, i.e., Turbo codes with the same component codes. While it proves that asymmetric Turbo codes with particular associations of component codes can achieve relatively good bit error rate (BER) performance in both “water-fall” and “error-floor” regions [13]-[15]. Thus it is necessary to identify asymmetric Turbo codes.

This paper aims to identify Turbo codes from a closed set based on the soft information, i.e., Log-Likelihood Ratio (LLR) accumulation and zero insertion decoding by taking advantage of LLR, instead of hard decision, is the proposed algorithm robust to noise. Moreover, this approach applies to both symmetric and asymmetric Turbo codes. In general, it has two stages: 1) identification of RSC1 by accumulating LLR of syndrome; 2) reconstruction of interleaver and identification of RSC2 by decoding based on zero insertion and accumulating LLR. The second stage consists of three steps: First, the bit stream is inserted by zeros and then decoded with BCJR algorithm. Second, the interleaver is recovered by bit-exclusive-or algorithm. At last, the input of the RSC2 is updated and the LLR of syndrome former is calculated to identify the encoder. Simulation results show that it identifies Turbo codes at signal-to-noise ratios (SNR) larger than 3.5dB.

The rest of this paper is organized as follows. Section II reviews Turbo codes and LLR algebra. Our proposed approach is detailed in Section III. Simulation results are demonstrated in Section IV. And Section V concludes this paper.

II. TURBO CODES AND LLR ALGEBRA

A. Turbo Codes

Turbo codes are composed of two recursive system convolution (RSC) encoders and one interleaver as illustrated in Figure. 1. The two RSC encoders are the same in symmetric Turbo codes, while for asymmetric Turbo codes they are not. Obviously blind identification for asymmetric Turbo codes is suitable for symmetric Turbo codes. Thus hereinafter Turbo codes refer to asymmetric Turbo codes.

For Turbo codes, the code-words consist of three parts: first, the information bit stream X is directly copied to the encoded output; then X is encoded by RSC1 to output the parity check bit stream Y ; at the same time X is encoded by RSC2 after permuted by the interleaver and the other parity check bit stream Z is output

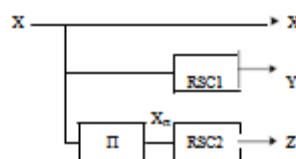


Figure 1: Turbo code

We assume that the generator matrices of the component encoders are respectively chosen from two sets of candidates Denoted by

$$\theta_1 = \{0_1^{(1)}, 0_1^{(2)}, \dots, 0_1^{(k)}\} \quad \text{and} \\ \theta_2 = \{0_2^{(1)}, 0_2^{(2)}, \dots, 0_2^{(k)}\}, \text{ where } 0_i^{(j)} \text{ stands for the generator matrices of the } i\text{-th encoder, } i=1, 2. \text{ The length}$$

Inter leaver N is assumed to be known. Therefore, the objective of our work is to identify which two generator matrices have been chosen and to recover the permutation of the inter leaver.

In this paper, we present to identify the parameters of Turbo codes based on LLR accumulation and zero insertion decoding. The proposed scheme is robust to noise since it takes use of the soft information than hard decision.

B. LLR Algebra

First let us introduce the basis of LLR algebra.

A priori L-value, which is a soft value measuring how high the probability of a binary random variable u being +1 is in comparison with that of u being -1 :

$$L_u(\mu) = \ln \frac{p_{u(\mu=+1)}}{p_{u(\mu=-1)}} \quad (1)$$

This is a priori information known before the result y caused by u becomes available. While the sign of LLR

$$\hat{\mu} = \text{sign}\{L_u(\mu)\} = \begin{cases} +1 & P_u(\mu = +1) > P_u(\mu = -1) \\ -1 & P_u(\mu = +1) < P_u(\mu = -1) \end{cases} \quad (2)$$

is a hard value denoting whether or not the probability of u being +1 is higher than that of u being -1, the magnitude of LLR is a soft value describing the reliability of the decision based on u . Conversely, $P_u(\mu = +1)$ and $P_u(\mu = -1)$ can be derived $L_u(\mu)$

$$P_u(\mu) = \frac{e^{(u+1)L(u)/2}}{1+e^{L(u)}} \quad (3)$$

$$P_u(\mu) = \begin{cases} e^{\frac{L(u)}{1+e^{L(u)}}} \\ \frac{1}{(1+e^{L(u)})} \end{cases} \quad (4)$$

III. BLIND IDENTIFICATION FOR TURBO CODES

In this paper, the received encoded bit stream is divided in to three branches: the information bits X, the parity check bits of RSC1 Y and the parity check bits of RSC2 Z. Our blind identification approach for Turbo codes has two stages. First, we identify RSC1 by LLR for syndrome prior probability algorithm. RSC2 is then identified by decoding based on zero insertion and LLR accumulation.

A. Identification for RSC1 Based on LLR Accumulation

The component code RSC1 is recursive systematic convolutional code.

For RSC1, given an encoder θ'_1 and its parity-check matrix $H_{\theta'_1}$, we can obtain:

$$H_{\theta'_1} c_v^0 = 0, \text{ if and only if } \theta = \theta'_1 \quad (5)$$

Where c_v stands for the code-word. Note that the elements of $H_{\theta'_1}$ are “0” or “1”, thus we can rewrite (5) as

$$c_{v,i1}^{\theta} \oplus c_{v,i2}^{\theta} \oplus c_{v,i3}^{\theta} \oplus \dots \oplus c_{v,ik}^{\theta} = 0 \quad (6)$$

If and only if $0=0'_1$.

For RSC1, its input bit stream X and its parity check bit stream Y are already known. Because of its systematic property, the encoded bits of RSC1 can be built up with X and Y.

Let $X = (x_1, x_2, \dots, x_N)$ denote the input bits and

$Y = (y_1, y_2, \dots, y_N)$ denote the parity check bits. Then the encoded bits of RSC1 can be shown as

$$out1 = (x_1, y_1, x_2, y_2, \dots, x_N, y_N) \quad (7)$$

According to [11][12], for the encoded bit stream, if the parity check relations satisfy (6), the syndrome posterior

Probability illustrated as follows would most probably take high values.

$$y_i \triangleq \bigoplus_{j=1}^p LLR(c_v / y_0) \quad (8)$$

Thus we can identify the encoder of RSC1. The generator matrix of RSC1 is chosen from candidate set θ_1 .

We Calculate $y_{0(k)} \triangleq \sum_{i=1}^k |y_i|$ for each $\theta_1^{(i)} \in \theta_1$. If $\theta_1^{(i)}$ is taken as the encoder, its $y_{\theta}(K)$ should be the maximal in all the accumulations.

B. Identification for RSC2 and Recovery of Interleaver

The key idea of LLR accumulation is the parity check relations in the code-words. However, for RSC2, the information bit stream X is permuted by the interleaver Π resulting the absence of the information bits for the code-words. Thus the LLR algorithm is no longer applicable. Figure.2 illustrates the structure of the interleaver Π and RSC2

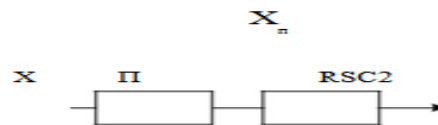


Figure 2: Interleaver and RSC2

The information bits denoted by $X = (x_1, x_2, \dots, x_N)$ are permuted by the interleaver with output denoted by $X_{\Pi} = (x_{\pi 1}, x_{\pi 2}, \dots, x_{\pi N})$. Then X_{Π} are encoded by RSC2 as

information bits to get the parity check bits

$$Z = (z_1, z_2, \dots, z_N)$$

To identify the encoder, we need to recover the missing information bits. Note that the intermediate variable X_{Π} is not only the output of the interleaver, but also the input for the encoder. If we could get x_{π} , we can easily recover the

interleaver and then identify the encoder. Therefore we firstly get the intermediate variable X_{Π} .

The encoder of RSC2 is chosen from a candidate set denoted by Θ_2 and the length of the interleaver is N .

The identification of RSC2 has three steps.

Step 1 Decoding based on zero insertion.

Step 2 Recovery interleaver based on Bit-Exclusive-Or.

Step 3 RSC2 identification based on LLR algorithm.

Step 1: Decoding Based on Zero Insertion

As mentioned before, in the context of RSC2, the information bits are out-of-order as a result of the interleaver, which makes it unable to generate the complete code-words sequence directly. Thus we need to recover the input information bits of RSC2 X_{Π} to identify the encoder by the

LLR algorithm. As the output of the interleaver Π , X_{Π} is a permutation of X . The only prior knowledge for the interleaver is its length N . Until now, there is no efficient way to recover X_{Π} under this condition. Therefore we have to recover X_{Π} through its another status which is the input

Information bits for RSC2. Fortunately, as for the encoder, the correspondence between the input information bits and the encoded output bits are one-to-one. Thus we can recover X_{Π} based on decoding the code-words of RSC2. However, the output of RSC2 is incomplete with its information bits missing. Thus the parity check bit stream Z is preprocessed by zero insertion to complete the code-words then decoded with BCJR algorithm to obtain X_{11} .

...

The parity check bit stream Z is preprocessed by zero insertion, which means we insert zeros instead of the information bits. Let $Z[z_1, z_2, \dots, z_N]$ denote the parity

check bits. We insert 0 to the corresponding information positions. For RSC2, the code-word sequence should be shown as

$$out2 [0, z_1, 0, z_2, \dots, 0, z_N]. \quad (9)$$

Obviously the encoded bit stream in (9) is of many different bits compared with the actual code-words. Thus the efficient decoding algorithm is required. Here we use BCJR algorithm [16]. It is a soft-input soft-output (SISO) algorithm based on posteriori probability of the information bits. It aims to find the maximal posterior probability of the

information bits, and performs well at high BER as it takes use of the soft information of the demodulated data rather than the hard decision.

BCJR algorithm needs two inputs: the complete encoded sequence arranged at intervals of information bits and parity check bits which we already obtained in (9), and the priori knowledge of the information bits. On the condition of BPSK modulation, 0/1 are mapping onto -1/+1. If the information bits are zeros, the priori probability will be set as 0.5 for BCJR. Thus the preliminary X_{Π} can be decoded based on (9) with BCJR algorithm.

Step 2: Recovery interleaver based on Bit-Elusive-Or

In step 1, the output of the interleaver X_π is obtained. Thus for the interleaver π , its input sequence X , output sequence X_π and its length n are already known. Thus the interleaver can be recovered based on bit-exclusive-or algorithm. This algorithm can be illustrated as follows:

Let matrix A and matrix B with N columns respectively denote the input and the output of the interleaver, where

$$A = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ x_{m1} & x_{m2} & \cdots & x_{mN} \end{bmatrix}$$

$$B = \begin{bmatrix} x_{\pi 11} & x_{\pi 12} & \cdots & x_{\pi 1N} \\ x_{\pi 21} & x_{\pi 22} & \cdots & x_{\pi 2N} \\ \cdots & \cdots & \cdots & \cdots \\ x_{\pi m1} & x_{\pi m2} & \cdots & x_{\pi mN} \end{bmatrix}$$

Take the column vector $x_i (1 \leq i \leq N)$ from A , then calculate the bit-exclusive-or with each column vector $x_{\pi j} (1 \leq j \leq N)$ of B . The results are saved in to an $N \times N$ matrix C , whose elements $c_{i,j}$ represent the different numbers of the i -th column of A and the j -th column of B .

The index of the minimum value of the i -th row of C stands for the mapping position for i . Then the interleaver Π can be reconstructed.

Step3: RSC2 Identification Based on LLR Algorithm

After step 2, the interleaver Π is recovered. Since its input information bit stream X is already known, its output bit stream X_Π can be updated, which is also the information bits of RSC2. Thus the code-words of RSC2 can be filled up by X_Π and Z . Compared with the code-words in step 1 obtained from zero insertion, the updated code-words are much more accurate. Based on this code-words, the RSC2 can be identified by the LLR accumulation.

IV. SIMULATION RESULTS

Simulations are shown in this section to evaluate the performance of our scheme. The generator matrices of RSC1 and RSC2 are both chosen from Θ , where

$\Theta = \{G_1, G_2, G_3, G_4, G_5\}$ are illustrated in Table I.

Table I: Five Encoders of Rsc1 and Rsc2

Encoder	Generator Matrix		
G1	$\begin{bmatrix} 1 & 0 & 1 & 1; 1 \\ & & & 1 \end{bmatrix}$	0	1]
G2	$\begin{bmatrix} 1 & 1 & 0 & \\ & 1; 1 & 1 \\ & & & 1 \end{bmatrix}$		1]
G3	$\begin{bmatrix} 1 & 0 & 0 & \\ & 1 & 1; 1 \\ & & & 1 \end{bmatrix}$		1 0 1]
G4	$\begin{bmatrix} 1 & 1 & 0 & \\ & 0 & 1; 1 \\ & & & 0 \end{bmatrix}$		1 1 1]
G5	$\begin{bmatrix} 1 & 1 & 0 & \\ & 0 & 1; 1 \\ & & & 1 \end{bmatrix}$		1 1 1]

At the transmitter, G_1 is chosen to be the encoder of RSC1 and G_2 is the encoder of RSC2. The interleaver is

randomly generated and its length N is 1784. Rate of the Turbo codes is $1/3$. The coded bit stream is modulated by BPSK, then transmitted over an additive white Gaussian noise (AWGN) channel. As illustrated in Section II, the received code-words stream is split into three parts including the information bit stream X , parity check bit stream of RSC1 Y and parity check bit stream of RSC2 Z .

A. Identification Result of RSC1

The encoded bit stream of RSC1 is made up with information bit stream X and parity bit stream Y . Thus the LLRs for syndrome a posteriori probabilities are directly calculated for each $\theta \in \Theta$ by (8).

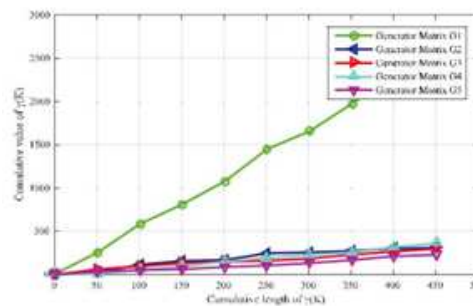


Figure 3: Comparison of Cumulative Metrics for RSC1 With All the Five Generator Matrices for SNR = 3.5 Db

Note that the accumulation differences between the true generator matrix and the other matrices are smaller for SNR=3.5dB than the one in identification of RSC2 introduces a lot of uncertain bits which leads to the inaccuracy of the interleaver recovery. At last, the Accumulation of LLR is based on code-words with higher BER. As a result, the differences between the right matrix and the other candidates are less.

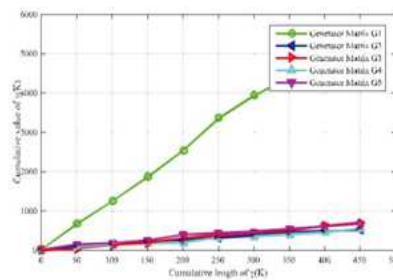


Figure 4. Comparison of Cumulative Metrics for RSC1 with All the Five Generator Matrices for SNR = 5 Db

Figure. 3 and Figure. 4 demonstrate the accumulations of LLR for five matrices for SNR=3.5dB and 5dB. We can easily find out that in both figures, the accumulation of generate matrix G_1 is much higher than the other four generator matrices. It indicates that G_1 is identified as the encoder of RSC1.

B. Identification Result of Rsc2

For identification of RSC2, the bit stream is firstly dealt with zero insertion and decoded with each $\theta \in \Theta$. Then based on X and all the X_{Π} , the interleaver Π can be recovered. After this, all the X_{Π} can be updated. At last, the LLRs are accumulated to identify the encoder.

Figure. 5 and Figure. 6 show the accumulations of LLR of all the five generator matrices for SNR=3.5dB and SNR=5dB. In both figures, the accumulation curves of $\gamma\theta(K)$ for matrix G2 are larger than the other four generator matrices. It indicates that the encoder of RSC2 is G2

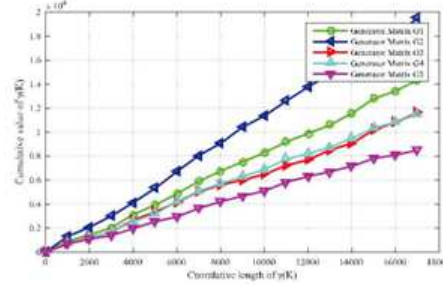


Figure 5: Comparison of Cumulative Metrics for RSC2 With All the Five Generate Matrices for SNR = 3.5 Db

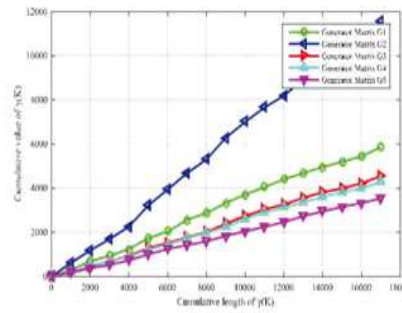


Figure 6: Comparison of Cumulative Metrics for RSC2 with All the Five Generate Matrices for SNR = 5 Db

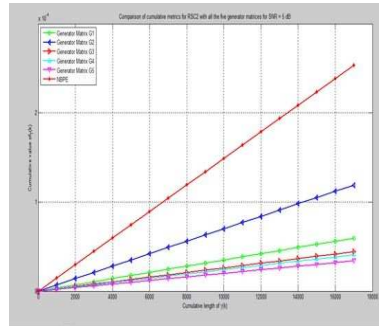


Figure 7: Novel blind parameter-estimation

The improvement for identification of RSC2 and the application for Turbo codes with various rates. we design a novel blind encoder parameter estimator for turbo codes. We propose to separate the feedback components from the forward path in a recursive convolution encoder so as to blindly estimate the parameters.

V. CONCLUSION

In this paper, we efficiently identify Turbo codes from closed-sets for AMC systems. The proposed method applies to both symmetric and asymmetric Turbo codes. For the identification of RSC1, since it is easy to get the code-words from the received bits, the accumulation of LLRs is directly used. For RSC2, the received bits are preprocessed with zero insertion and decoded with BCJR algorithm to obtain its encoded bits at first. The interleaver is recovered based on

the encoded bits and then the code-words are updated to accumulate the LLRs. Simulation results show that our scheme can blindly identify Turbo codes at SNR larger than 3.5dB. The improvement for identification of RSC2 and the application for Turbo codes with various rates. we design a novel blind encoder parameter estimator for turbo codes. In this scheme, we propose to separate the feedback components from the forward path in a recursive convolution encoder so as to blindly estimate the parameters. The simulations demonstrate the average estimation performance can lead to more than 95% accuracy for the channel condition with the signal-to noise ratio at 5 dB. Once the encoder parameters are blindly estimated, the corresponding decoder can be implemented to retrieve the original information sequences. Continued work on this method may include the improvement for identification of RSC2 and the application for Turbo codes with various rates.

REFERENCES

1. Goldsmith A J, Chua S G. Adaptive coded modulation for fading channels. *Communications, IEEE Transactions on*, 1998, 46(5): 595-602.
2. Goeckel, Dennis L. "Adaptive coding for time-varying channels using outdated fading estimates." *Communications, IEEE Transactions on* 47.6 (1999): 844-855.
3. Priti Subramaniam & Rajeshree D. Raut, *Performance Analysis of Turbo Coded OFDM: Subjected to Different Channels*, *International Journal of Computer Science Engineering and Information Technology Research (IJCSSEITR)*, Volume 4, Issue 1, January - February 2014, pp. 25-34
4. Xia T, Wu H C, "Novel blind identification of LDPC codes using average LLR of syndrome a posteriori probability," *IEEE Transactions on Signal Processing*, vol.62, no.3, pp.632-640, 2014.
5. C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *ICC'93 Proc, Geneva, 1993*, pp.1064-1070.
6. Divsalar D, Pollara F. "Turbo codes for deep-space communications," *JPL TDA Progress Report 42-120*. Pasadena Jet Propulsion Lab, 1995.
7. WCDMA proposal: 3GPP proposals TS S1.12 V2.0.0, TS 25.212. V2.2.0
8. Naseri A, Azmoon O, Fazeli S, "Blind recognition algorithm of Turbo codes for communication intelligence systems," *Int. J. Comput. Sci. Issues*, 8(6), 2011.
9. Debessu Y G, Wu H C, Jiang H, "Novel blind encoder parameter estimation for turbo codes," *IEEE Communications Letters*, vol.16, no.12, pp.1917-1920, 2012.
10. Cluzeau M, Finiasz M, Tillich J P, "Methods for the reconstruction of parallel turbo codes," *arXiv preprint arXiv:1006.0259*, 2010.
11. Wu Hengzhou, Luo Xiaobin, Liu Jie. "Research on blind recognition of Turbo codes," *Radion Engineering*, 2015, 45(5), pp.24-27.
12. P. Yu, J. Li, and H. Peng, "A least square method for parameter estimation of RSC sub-codes of turbo codes," *IEEE Commun. Lett.*, vol. 18, no. 4, pp. 644-647, Apr. 2014.
13. Li Xiaotian, Zhang Runsheng, Li Yanbin, "Research on the recognition algorithm of Turbo codes on trellis termination," *Journal of XIDIAN University*, 2013, vol.40, no.4, pp.161-166.

14. Takeshita O Y, Collins O M, Massey P C, et al. "Asymmetric turbo-codes," *IEEE International Symposium on Information Theory*.
15. *INSTITUTE OF ELECTRICAL ENGINEERS INC (IEEE)*, 1998: 179-179.
16. Takeshita O Y, Collins O M, Massey P C, et al. "A note on asymmetric turbo-codes," *IEEE Communications letters*, 1999, 3(3): 69-71.MLA
17. Massey P C, Costello Jr D J, "New developments in asymmetric turbo codes," in *Proc. of the 2nd International Symposium on Turbo Codes and Related Topics*, 2000, pp.93-100.
18. Moosavi R, Larsson E G, "A fast scheme for blind identification of channel codes," *54th Annual IEEE Global Telecommunications Conference:" Energizing Global Communications"*, *GLOBECOM 2011; Houston, TX; United States*. 2011, pp.1-5.